IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant: Marek Musial

Serial No.: 09/602,418

Filed: June 23, 2000

For: PROCESS FOR LEARNING THE BASIC FINITE AUTOMATON OF A PROTOCOL IMPLEMENTATION

Examiner: Min Jung

Art Unit: 2616

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Appeal Brief in Accordance With 37 C.F.R. § 41.37

Dear Sir:

This is an appeal from the Examiner's final rejection of the above-identified application dated February 21, 2007.

Please charge the one hundred twenty dollar ($120) fee for the one month extension of time under 37 C.F.R. § 1.17(a)(1) to Deposit Account 20-0352.

No additional fee is believed due. However, if an additional fee is due please charge that fee to Deposit Account 20-0352.

Table of Contents

Real Party in Interest

The real party in interest in this case is Appellant's assignee, Tektronix, Inc., an Oregon corporation.

Related Appeals and Interferences

There are no prior and pending appeals, interferences or judicial proceedings known to Appellant, Appellant's legal representative or assignee which may be related to, directly affect or have a bearing on the Board's decision in this appeal.

Status of Claims

Claims 2-14 and 16 stand rejected under 35 U.S.C. § 101 and are being appealed.

Claims 1 and 15 are cancelled.

Status of Amendments

No amendments have been submitted by Appellant after the Examiner's final rejection.

Summary of Claimed Subject Matter

Independent claim 2 is a method of learning a finite automaton (page 1, line 21 – page 2, lines 5) of a protocol implementation of a communication system, comprising the steps of (page 4, lines 19-23): a) grouping times within an example communication (page 9, line 5-16) including Protocol Data Unit (PDU) types together as equivalence classes; and b) using the equivalence classes as states of the finite automaton. The grouping step comprises the steps of (page 5, lines 3-7): calculating a similarity value between every two times within the example communication to form a similarity matrix, the similarity values being dependent on the length of the PDU type sequence which is coincident for and surrounds both times; and forming the equivalence classes from the similarity matrix.

Independent claim 5 is a method of learning a finite automaton (page 1, line 21 – page 2, lines 5) of a protocol implementation of a communication system comprising the steps of (page 4, lines 19-23): a) grouping times within an example communication (page 9, line 5-16) including PDU (Protocol Data Unit) types together as equivalence classes; and b) using the equivalence classes as states of the finite automaton. The using step comprises the step of (page 5, lines 15-21) entering each PDU of the example communication as a state transition of the finite automaton, the state transition being a transition from the state whose equivalence class includes the time immediately prior to the PDU to the state whose equivalence class includes the time immediately after the PDU marked with the PDU type.

Independent claim 7 is a method of learning a finite automaton (page 1, line 21 – page 2, lines 5) of a protocol implementation of a communication system using an example communic-

ation (page 9, line 5-16), the finite automaton having basic protocol states and the state transitions of the finite automaton being marked with an appropriate Protocol Data Unit (PDU) type, comprising the steps of: a) defining times in the example communication between every two PDUs which occur in sequence; b) calculating a similarity value between every two times as defined in a) to form a similarity matrix, which similarity value indicates the sum of the number of PDU types coincident for and surrounding both times (page 5, lines 3-7); c) transforming the similarity matrix to an equivalence matrix by means of a lower threshold for the similarity values calculated according to b), such that two times fulfill an equivalence relation for an equivalence matrix if the similarity values between these two times is larger than or equal to the lower threshold (page5, lines 8-12); d) forming a transitive hull for the equivalence matrix defined according to c), the transitive hull constituting equivalence classes on times according to a) (page 5, lines 12-14); e) defining each equivalence class of the equivalence relation according to d) as a state of the finite automaton; f) entering the PDUs of the example communication as state transitions of the finite automaton, the state transitions being a transition from the state whose equivalence class according to e) includes the time immediately prior to the PDU in question to the state whose equivalence class according to e) includes the time immediately after the PDU in question marked with the PDU type of the PDU in question, with transitions that are identical as far as starting and sequential states and PDU type are concerned being only entered once. (page 5, lines 15-21)

Independent claim 9 is a method of learning arithmetic classification rules (page 6, line 5) for features of a finite automaton (page 1, line 21 – page 2, lines 5) of a protocol implementation of a communication system from a training set (page 9, line 5-16) having positive examples

(page 6, lines 9-14) comprising the steps of: a) forming derived features from the training set on the basis of statistical measures in the form of arithmetic terms; and b) formulating logic conditions on numerical values of the group consisting of the features from the training set and the derived features (page 6, lines 7-9); the training set being an example communication composed of Protocol Data Units (PDUs) of a protocol machine and the logic conditions being the rules for the numerical PDU field contents of a sequence of PDUs.

Independent claim 16 is a method of learning arithmetic classification rules (page 6, line 5) for features of a finite automaton (page 1, line 21 – page 2, lines 5) of a protocol implementation of a communication system from a training set (page 9, line 5-16) having exclusively positive examples (page 6, lines 9-14), the method being used for learning rules for numerical Protocol Data Unit (PDU) field contents of a sequence of PDUs which correspond to a specific partial path in a finite automaton of protocol basic states and PDU types, comprising the steps of (page 6, line 15 – page 7, line 12): a) interpreting each component of a feature vector as the expression of an attribute, with the number of attributes present at the beginning corresponding to the dimension of the feature vector; b) forming new, derived attributes for each possible attribute pair on the basis of correlation and regression coefficients on the training set, with the value of a derived attribute for each feature vector being calculated from already present attribute values of the feature vector, namely as a sum, product, quotient or difference of two already present attributes, or as a product of a present attribute and a constant, in the case of a calculation from two present attributes, the attributes considered in the calculation being selected for maximum correlation with a third attribute and the arithmetic operation for maximum correlation of the derived attribute with the same third attribute, and in the case of a

multiplication with a constant, an attribute with a particularly high correlation with a second attribute being multiplied by the linear regression coefficient of the attribute pair in such a manner that the resulting derived attribute corresponds numerically to the said second attribute, if possible; c) deriving conspicuous accumulations of the values of an original attribute or an attribute according to b) that are detected in a numerical value or within a numerical interval, a conspicuous accumulation being defined in that it maximizes the quotient of the width of the smaller one of the two gaps immediately adjacent to the numerical interval in which there are no values of the attribute in question, and the width of the largest gap within the numerical interval in which there are no values of the attribute in question; d) forming clauses based on conspicuous accumulations as defined in c), said clauses each formulating a logic condition for selecting those examples from the training set whose attribute values of a certain attribute are within a time interval determined according to the characteristics of the associated conspicuous accumulation as defined in c), with each clause being capable of representing a conjunction of plural such selection criteria for different attributes; e) constructing a selection of the clauses according to d) for characterizing the entire training set such that all elements, if possible, of the training set are selected by at least one of the clauses, and as many as possible of them by exactly one clause.

Grounds of Rejection to be Reviewed on Appeal

Whether claims 2-14 and 16 are unpatentable under 35 U.S.C. § 101.

Argument

The Examiner rejected claims 2-4 and 16 under 35 U.S.C. § 101 alleging that the claimed invention lacks patentable utility. Specifically, the Examiner writes that the claims are abstract ideas because they are merely data manipulation lacking physical application which does not produce a "useful, concrete, and tangible result." The Examiner acknowledges that the utility of the claimed invention is described in Appellant's disclosure, but rejects the claims nonetheless because that utility is not expressly recited in the claims.

Appellant asserts that claims 2-4 and 16 satisfy 35 U.S.C. § 101 on two independent grounds: First, claims 2-4 and 16 satisfy 35 U.S.C. § 101 because they are a practical application of an abstract idea in that their results "relate to" a useful, concrete, and tangible thing: a finite state machine implementing a protocol on a communication network. *Arrhythmia Research Technology Inc. v. Corazonix Corp.,* 958 F.2d 1053, 1059 (Fed.Cir. 1992). Second, claims 2-4 and 16 satisfy 35 U.S.C. § 101 because they have a "well-established utility," i.e., (i) a person of ordinary skill in the art of protocol analysis would immediately appreciate that Appellant's invention is useful for protocol testing, and (ii) this utility is specific, substantial, and credible. (See MPEP § 2107(II)) Under either theory, there is no requirement in 35 U.S.C. § 101 or in the case law that this utility be expressly recited in the claims.

Claims 2-14 and 16 Produce a "Useful, Concrete, and Tangible Result"

Independent claim 2 satisfies 35 U.S.C. § 101 because it produces a "useful, concrete, and tangible result." The result of claim 2 is "knowledge of a finite automaton of a protocol implementation of a communication system," or in other words, knowledge of the states and state transitions (page 5, line 2) of a state machine (page 1, line 22) implementing a network

communication protocol. Specifically, given a measured "example communication," i.e., a training sequence exchanged between two communication partners (page 9, lines 7-8), Appellant's claimed invention teaches methods of analyzing that example communication in order to automatically derive the states and state transitions of the underlying finite state machine, and thus, all information necessary for a complete protocol test algorithm. (page 4, lines 4-9)

Such a result is "useful" because it facilitates the troubleshooting and validation of communication systems. Prior art protocol analyzers require a considerable amount of human design work (page 4, line 3), particularly when the protocol description does not exist in a formal machine-compatible or machine-readable form (page 4, line 15). Therefore, automatically deriving a finite automaton from an example communication, which provides all necessary information for a complete protocol test algorithm, is "specific, substantial, and credible" utility. (See MPEP § 2106(IV)(C)(2)(2)(a))

Such a result is "concrete" because it is "reproducible" and "predictable." (See MPEP § 2106(IV)(C)(2)(2)(c)) That is, one of ordinary skill in the art of protocol analysis recognizes that the steps of "grouping . . . using . . . calculating . . . and forming . . ." are deterministic mathematical operations performed on digital data, and thus the result is "reproducible" and "predictable."

Such a result is "tangible" because it constitutes a practical application of an abstract idea (See MPEP § 2106(IV)(C)(2)(2)(b)) in that it "relates to" a useful, concrete or tangible thing: a finite state machine implementing a protocol on a communication network. See *Arrhythmia Research Technology Inc. v. Corazonix Corp.,* 958 F.2d 1053 (Fed.Cir. 1992), holding that the mathematical transformation of electrocardiograph signals from a patient's heartbeat constituted

a practical application of an abstract idea because it "related to" a useful, concrete or tangible thing: the condition of the patient's heart. Similarly, Appellant's claimed invention mathematically transforms a measured communication signal and constitutes a practical application of an abstract idea because it "relates to" a finite state machine implementing a protocol on a communication network. This is certainly a "real-world result" (See MPEP § 2106(IV)(C)(2)(2)(b)), not a mere "abstract idea." *In re Alappat,* 33 F.3d 1526, 1543 (Fed.Cir. 1994).

For these reasons, claim 2 produces a "useful, concrete, and tangible result." Therefore, Appellant requests that the rejection of claim 2 under 35 U.S.C. § 101 be reversed.

For all the same reasons, claims 3-14 and 16 also produce a "useful, concrete, and tangible result." Therefore, Appellant requests that the rejection of claims 3-14 and 16 under 35 U.S.C. § 101 be reversed.

<u>Claims 2-14 and 16 have a "Well-Established Utility"</u>

Independent claim 2 satisfies 35 U.S.C. § 101 because it has a "well-established utility," i.e., (i) a person of ordinary skill in the art of protocol analysis would immediately appreciate that Appellant's invention is useful for protocol analysis, and (ii) this utility is specific, substantial, and credible. (See MPEP § 2107(II)(A)(3).)

Those of ordinary skill in the art of protocol analysis are very familiar with the challenges of developing test algorithms for communication protocols. (page 3, line 23 – page 4, line 18) In particular, prior art protocol analyzers require a considerable amount of human design work (page 4, line 3), particularly when the protocol description does not exist in a formal machine-compatible or machine-readable form (page 4, lines 15). Thus, one of ordinary skill in the art of

protocol analysis would immediately appreciate the "specific, substantial, and credible" utility of a machine-based learning process which allows test algorithms to be developed almost without any human design work (page 4, line 5) and without a formal, machine-compatible or machine-readable specification.

For these reasons, claim 2 has a "well-established utility." Therefore, Appellant requests that the rejection of claim 2 under 35 U.S.C. § 101 be reversed.

For all the same reasons, claims 3-14 and 16 also have a "well-established utility." Therefore, Appellant requests that the rejection of claims 3-14 and 16 under 35 U.S.C. § 101 be reversed.

Claims 2-14 and 16 Need Not Expressly Recite a Practical Application

The Examiner acknowledges that the utility of claims 2-14 and 16 is described in Appellant's disclosure, but rejects them nonetheless because that utility is not expressly recited in the claim language. Specifically, the Examiner writes that the end result of claims 2-14 and 16 is merely a number without a practical application expressly recited in the claim. Appellant understands that MPEP § 2106(IV)(C)(2)(2)(a) gives the Examiner the authority to require Appellant to specifically recite a practical application in the claim, however Appellant asserts that this authority is unsupported by 35 U.S.C. § 101 or the case law. As MPEP § 2106(I) itself concedes: "These Guidelines do not constitute substantive rulemaking and hence do not have the force and effect of law."

According to the case law, it is well-established that a claim may satisfy 35 U.S.C. § 101 without expressly reciting a practical application. See, for example, *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 149 F.3d 1368 (Fed.Cir. 1998), holding that the end

result, a share price, i.e. a number, is useful because it is "relied upon . . . in subsequent trades," even though the reliance and subsequent trades are not expressly recited in the claim. Also see *Arrhythmia*, holding that the end result, an indication of energy in a signal, i.e., a number, constitutes a practical application of an abstract idea because it "relates to" a useful, concrete or tangible thing—the condition of a patient's heart, even though the patient's heart is not expressly recited in the claim.

In fact, in terms of form, Appellant's independent claim 2 is nearly identical to the claim at issue in *Arrhythmia*, both of which are listed below, with underlining added for the sake of comparison: Both claims are methods of analyzing measured signals in order to learn information; in neither case is the practical application of that information expressly recited in the claim.

> Appellant's claim 2: "A method of learning a finite automaton of a protocol implementation of a communication system comprising the steps of:
> a) grouping times within an example communication including PDU (Protocol Data Unit) types together as equivalence classes; and
> b) using the equivalence classes as states of the finite automaton;
> the grouping step comprising the steps of:
> calculating a similarity value between every two times within the example communication to form a similarity matrix, the similarity values being dependent on the length of the PDU type sequence which is coincident for and surrounds both times; and
> forming the equivalence classes from the similarity matrix."

> *Arrhythmia* claim 1: "A method for analyzing electrocardiograph signals to determine the presence or absence of a predetermined level of high frequency energy in the late QRS signal, comprising the steps of:
> converting a series of QRS signals to time segments, each segment having a digital value equivalent to the analog value of said signals at said time;
> applying a portion of said time segments in reverse time order to high pass filter means;
> determining an arithmetic value of the amplitude of the output of said filter; and
> comparing said value with said predetermined level."

The United States Court of Appeals for the Federal Circuit approved of the claim at issue in *Arrhythmia*, writing: "These input signals are not abstractions; they are related to the patient's heart function. . . . The resultant output is not an abstract number, but is a signal related to the patient's heart activity." *Arrhythmia Research Technology, Inc. v. Corazonix Corp.*, 958 F.2d 1053, 1059 (Fed.Cir., 1992). Similarly, Appellant's example communication is not an abstraction, but it is related to the operation of a finite state machine on a communication network. Appellant's resultant output is not an abstract number, but it is related to the activity of the finite state machine.

For these reasons, claims 2-14 and 16 need not expressly recite a practical application in order to satisfy 35 U.S.C. § 101. Therefore, Appellant requests that the rejection of claims 2-14 and 16 under 35 U.S.C. § 101 be reversed.

Conclusion

For all these reasons, Appellant requests that the Examiner's rejection of claims 2-14 and 16 be reversed, and that this case be passed on to issuance.

Respectfully submitted,

Marek Musial

By: /Michael A. Nelson/

Michael A. Nelson
Reg. No. 59,450
(503) 627-1785 (Voice)
(503) 627-7119 (Fax)

August 16, 2007

Tektronix, Inc.
P.O. Box 500
Delivery Station 50-LAW
Beaverton, OR 97077

Claims Appendix

1. (Cancelled)

2. A method of learning a finite automaton of a protocol implementation of a communication system comprising the steps of:

a) grouping times within an example communication including PDU (**P**rotocol **D**ata **U**nit) types together as equivalence classes; and

b) using the equivalence classes as states of the finite automaton;

the grouping step comprising the steps of:

calculating a similarity value between every two times within the example communication to form a similarity matrix, the similarity values being dependent on the length of the PDU type sequence which is coincident for and surrounds both times; and

forming the equivalence classes from the similarity matrix.

3. The method as claimed in claim 2 wherein the forming step comprises the step of transforming the similarity matrix into an equivalence matrix by means of a lower threshold for the similarity values such that the similarity values are converted into states and the times are grouped together by state to form the equivalence classes.

4. The method as recited in claim 2 wherein the forming step comprises the steps of:

forming a transitive hull for the similarity matrix between two times within the example communication to calculate an equivalence relation; and

obtaining the equivalence classes from the equivalence relation.

5. A method of learning a finite automaton of a protocol implementation of a communication system comprising the steps of:

a) grouping times within an example communication including PDU (**P**rotocol **D**ata **U**nit) types together as equivalence classes; and

b) using the equivalence classes as states of the finite automaton;

the using step comprising the step of entering each PDU (Protocol Data Unit) of the example communication as a state transition of the finite automaton, the state transition being a transition from the state whose equivalence class includes the time immediately prior to the PDU to the state whose equivalence class includes the time immediately after the PDU marked with the PDU type.

6. The method as recited in claim 5 further comprising the step of performing the preceding steps several times for overlapping partial sections of the example communication, with the similarity matrix of two overlapping partial sections each being united to form the similarity matrix for the example communication.

7. A method of learning a finite automaton of a protocol implementation of a communication system using an example communication, the finite automaton having basic protocol states and the state transitions of the finite automaton being marked with an appropriate Protocol Data Unit (PDU) type, comprising the steps of:

a) defining times in the example communication between every two PDUs which occur in sequence;

b) calculating a similarity value between every two times as defined in a) to form a similarity matrix, which similarity value indicates the sum of the number of PDU types coincident for and surrounding both times;

c) transforming the similarity matrix to an equivalence matrix by means of a lower threshold for the similarity values calculated according to b), such that two times fulfill an equivalence relation for an equivalence matrix if the similarity values between these two times is larger than or equal to the lower threshold;

d) forming a transitive hull for the equivalence matrix defined according to c), the transitive hull constituting equivalence classes on times according to a);

e) defining each equivalence class of the equivalence relation according to d) as a state of the finite automaton;

f) entering the PDUs of the example communication as state transitions of the finite automaton, the state transitions being a transition from the state whose equivalence class according to e) includes the time immediately prior to the PDU in question to the state whose equivalence class according to e) includes the time immediately after the PDU in question marked with the PDU type of the PDU in question, with transitions that are identical as far as starting and sequential states and PDU type are concerned being only entered once.

8. The method as recited in claim 7 wherein steps a) to f) are performed several times for overlapping partial sections of the example communication, with the equivalence matrices according to c) of two overlapping partial sections each being united and the state-forming equivalence matrix being calculated in analogy to d) by means of the union of the equivalence matrices.

9. A method of learning arithmetic classification rules for features of a finite automaton of a protocol implementation of a communication system from a training set having positive examples comprising the steps of:

a) forming derived features from the training set on the basis of statistical measures in the form of arithmetic terms; and

b) formulating logic conditions on numerical values of the group consisting of the features from the training set and the derived features;

the training set being an example communication composed of Protocol Data Units (PDUs) of a protocol machine and the logic conditions being the rules for the numerical PDU field contents of a sequence of PDUs.

10. The method as recited in claim 9 wherein the forming step comprises the step of forming the derived features on the basis of correlation and regression coefficients on the training set for each possible pair of features, with the value of the derived feature being calculated from two features from the training set or from one feature from the training set and a constant.

11. The method as recited in claim 10 wherein the formulating step comprises the step of taking the conspicuous accumulations of the values of a feature from the training set or a derived feature in a numerical value or within a numerical interval into consideration to establish the logic conditions.

12. The method as recited in claim 11 wherein the conspicuous accumulation is defined in that it maximizes the quotient of the width of the smaller one of two gaps immediately adjacent to the numerical interval in which there are no values of the feature in question, and the width of the largest gap within the numerical interval in which there are no values of the feature in question.

13. The method as recited in claim 12 further comprising the step of:

constructing plural subclasses of the training set by organizing the logic conditions in a disjunction of clauses, with one clause constituting a conjunction of one or plural logic condition(s) and describing a subclass each of said training set.

14. The method as recited in claim 13 further comprising the step of conducting a selection of the constructed clauses for characterizing the entire training set such that all elements, if possible, of the training set are selected by at least one of the clauses, and as many as possible of them by exactly one clause.

15. (Cancelled)

16. A method of learning arithmetic classification rules for features of a finite automaton of a protocol implementation of a communication system from a training set having exclusively positive examples, the method being used for learning rules for numerical Protocol Data Unit (PDU) field contents of a sequence of PDUs which correspond to a specific partial path in a finite automaton of protocol basic states and PDU types, comprising the steps of:

a) interpreting each component of a feature vector as the expression of an attribute, with the number of attributes present at the beginning corresponding to the dimension of the feature vector;

b) forming new, derived attributes for each possible attribute pair on the basis of correlation and regression coefficients on the training set, with the value of a derived attribute for each feature vector being calculated from already present attribute values of the feature vector, namely as a sum, product, quotient or difference of two already present attributes, or as a product of a present attribute and a constant, in the case of a calculation from two present attributes, the attributes considered in the calculation being selected for maximum correlation with a third attribute and the arithmetic operation for maximum correlation of the derived attribute with the same third attribute, and in the case of a multiplication with a constant, an attribute with a particularly high correlation with a second attribute being multiplied by the linear regression coefficient of the attribute pair in such a manner that the resulting derived attribute corresponds numerically to the said second attribute, if possible;

c) deriving conspicuous accumulations of the values of an original attribute or an attribute according to b) that are detected in a numerical value or within a numerical interval, a conspicuous accumulation being defined in that it maximizes the quotient of the width of the smaller one of the two gaps immediately adjacent to the numerical interval in which there are no values of the attribute in question, and the width of the largest gap within the numerical interval in which there are no values of the attribute in question;

d) forming clauses based on conspicuous accumulations as defined in c), said clauses each formulating a logic condition for selecting those examples from the training set whose attribute values of a certain attribute are within a time interval determined according to the

characteristics of the associated conspicuous accumulation as defined in c), with each clause being capable of representing a conjunction of plural such selection criteria for different attributes;

e) constructing a selection of the clauses according to d) for characterizing the entire training set such that all elements, if possible, of the training set are selected by at least one of the clauses, and as many as possible of them by exactly one clause.

Evidence Appendix

No evidence was submitted pursuant to 37 C.F.R. §§ 1.130, 1.131 or 1.132, and no other evidence was entered by the Examiner.

Related Proceedings Appendix

There are no related proceedings identified in this Brief.